

## Method for Device-to-Device Pervasive Digital Output

### Technical Field of invention

**[0001]** Present invention relates to device synchronization and digital output and, in particular, to providing pervasive output for information apparatuses.

### Summary of the Invention

**[0002]** An Information apparatus refers to both stationary computers and mobile computing devices (pervasive devices). Examples of information apparatuses include without limitation desktop computers, laptop computers, palmtop (hand-held) computers, personal digital assistants (PDAs), Internet-enabled cellular phones, smart phones, pagers, Internet appliances, e-books, digital pads, Web pads, and digital capturing devices (e.g., digital cameras and video cameras). An output device 140 may include a fax machine, printer, copier, image or video display device, projector, and an audio output device.

**[0003]** For simplicity and convenience, hereafter, the following description may refer to an output device as a printer and an output process as printing. However, it should be understood that the term printer and printing is used as a specific example to simplify description or may be one exemplary embodiment. The reference to printing used here is intended to be applied or extended to the larger scope and definition of output devices and should not be construed as restricting the scope and practice of present invention.

**[0004]** Fueled by ever-increasing bandwidth and processing power and ever-increasing numbers of wireless mobile devices and available software applications for pervasive devices, millions of users are or will be creating, downloading, and transmitting content and information using their pervasive computing devices. Unfortunately, the small display screen of a pervasive device may not provide a good viewing

experience. As a result, there is a need to allow users to easily output content and information from their pervasive computing devices to any output device. People need to output directly and conveniently from their pervasive information apparatus, without depending on synchronizing with a stationary computer (e.g., desktop personal computer) for printing, as an example.

[0005] To illustrate, an information worker at an airport receiving e-mail in his hand-held computer may want to walk up to a nearby printer or fax machine to have his e-mail printed. In addition, the mobile worker may also want to print a copy of his to-do list, appointment book, business card, and his flight schedule from his mobile device. As another example, a traveler stopping at a convenience store, gas station, or kiosks may want to print out a copy of the map he just downloaded from the Internet using his Internet enable wireless phone (information apparatus). In still another example, an individual who just finished a stock trade or performed an e-commerce transaction using his PDA (information apparatus) may want to print out a copy of his transaction confirmation, invoice or receipt. In yet another example, a user who takes a picture with a digital camera may want to easily print it out to a nearby printer.

[0006] Conventionally, an output device (e.g., a printer) is connected to an information apparatus via a wired connection such as a cable line. A wireless connection is also possible by using, for example, radio communication or infrared communication. Regardless of whether the connection is wired or wireless, a user must first install in the information apparatus a printer driver corresponding to a particular printer model and make. Using a device-dependent or specific driver, the information apparatus may process output content or a digital document into the printer input space. The printer input space corresponds to the type of input that a printer understands (herein referred to as print data). For example, the printer input space or print data may include a printer-

specific input format (e.g., image, graphics, file, data format), encoding, page description language, markup language, instructions, protocols or data that can be understood or used by a particular printer make and model.

**[0007]** Print data may be proprietary or a published standard or a combination of the two. A printer input space or print data is therefore, in general, device dependent. Different printer models may specify their own input, designed or adopted for optimal operation by the printer manufacturer according to a specification. Consequently, different printers usually require use of specific printer drivers for accurate printing. A device driver (printer driver in this example) may control, manage, communicate, and output print data to a printer. Sometimes, instead of using a printer or device driver, the device driving feature may be included as part of an application software. The application in this example may process the document and communicate directly with an output device without utilizing a device driver. Print data is an output data sent to a printer. However, the term print data used here is intended to be applied or extended to the larger scope and definition of output data which applies to any output devices including display, projection and audio devices, and this should not be construed as restricting the scope and practice of present invention.

**[0008]** Installation of a printer driver or application may be accomplished by, for example, manual installation using a CD or floppy disk supplied by the printer manufacturer. Or alternatively, a user may be able to download that particular driver or application from a network. For a home or office user, this installation process may take anywhere from several minutes to several hours depending on the type of driver and user's sophistication level with computing devices and networks. Even with plug-and-play driver installation, the user is still required to execute a multi-step process for each printer. This installation and configuration process adds a degree of complexity and work to end-users who may

otherwise spend their time doing other productive or enjoyable work. Moreover, many unsophisticated users may be discouraged from adding new peripherals (e.g., printers, scanners, etc.) to their home computers or networks to avoid the inconvenience of installation and configuration. Therefore, there is a need to provide a way for a user to more conveniently or easily output digital content to an output device without the inconvenience of manually finding and installing new device drivers or printer drivers. Installation of printer drivers, if needed, should be more automatic and transparent to end-users.

[0009] In addition, conventional printing methods may pose significantly higher challenges and difficulties for mobile device users than for home and office computer users. The requirement for pre-installation of a device-dependent driver is in conflict with the concept of mobile (pervasive) computing and output. For example, a mobile user may want to print e-mail, PowerPoint® presentation documents, web pages, or other documents in an airplane or at an airport, gas station, convenience store, kiosk, hotel, conference room, office, home, etc. It is highly unlikely that the user would find at any of these locations a printer of the same make and model as is at the user's base station. Therefore, the user may have to install and configure a printer driver each time at each of these locations before printing. Alternatively, it is also not a viable option usually to pre-install all of the possible hundreds, or even thousands, of printer drivers available to the user's information apparatus. A mobile information device usually has limited memory and storage capacity. As a consequence, the user would currently be required to install and configure a printer driver each time at each such different location before printing. Moreover, the user may not want to be bothered with looking for a driver or downloading it and installing it just to print out one page of email at the airport. This is certainly an undesirable and discouraging process to promote pervasive or mobile computing. Therefore, a more convenient or automated printing solution is needed

so that a user can simply walk up to a printer and easily print a digital document without having to install or pre-install a particular printer driver.

**[0010]** Network printing may partially solve the above problem in a local area network (LAN). Network printing was first motivated to allow people to share printing resources within a network. Printers are often connected to the network using a network adapter or network card. Print servers may be used to manage print jobs (such as spooling). Different clients may send print jobs to the print server for queuing, prioritizing and managing. The driver of each networked printer may be stored in the network (for example, in an application server) and available for clients to download.

**[0011]** The above configuration is often used in a local area network (LAN) such as in an office building. An office worker with a mobile or static computing device connected to this network may be able to send a print job to any printer that is connected to the same network. He either has the necessary printer drivers pre-installed in his device or he can download a driver from the LAN before printing to a particular printer. But once outside of the office building, or meaning outside of his wired or wireless network coverage, the office worker can no longer print to any printer he or she desires.

**[0012]** In one instance, it has been argued that the idea of local network printing can be further extended to the Internet as the larger and public network. In an ideal situation, every printer and every computing device (information apparatus) is connected to the same network, for example the Internet. But in reality, extending network printing to a big super-network or the Internet is still unavailable.

**[0013]** Nevertheless, one drawback of network printing may be higher cost. To connect the output devices and information apparatus to the same network, additional hardware or software may need to be installed. And therefore, the costs of output devices may be raised substantially. In addition to hardware and software costs there are also service

FOUO - 149101

charges to maintain the connectivity to a large network such as the Internet. Furthermore, a large permanent network infrastructure would need to be built and made available to each location.

**[0014]** In addition to costs, another drawback of network printing is reliability. Network printing requires perfect function in any part of the chain of network hardware, software and services. If any component of the link is down or not fully functional, for any reason, printing service may be unavailable, interrupted or result in inaccuracies.

**[0015]** Finally, another drawback of network printing is the issue of security. A user does not feel secure if confidential documents or information must pass through the entire network before reaching the printer immediately in front of him or her. There may be the fear that the confidential document could be intercepted, viewed, copied or stored within any segment of the connected network. There can also be the fear that the document could actually be printed to a different printer instead of the intended one. This problem may be caused by, for example, errors in directory service software, a software component that is not up-to-date or corrupted, inconsistent or incompatible software and/or hardware versions, or any combination of human, software, network, or hardware errors. There is also risk when downloading a software component (e.g., printer driver) from the network to an information apparatus. This software component maybe damaged, corrupted or carrying a malicious attack or virus or otherwise modified by hackers. A damaged software component can disrupt or corrupt the user's information apparatus. Finally, some internal corporate or local networks may reject unknown documents trying to reach their printers and other output devices protected within a firewall. It is therefore more desirable if the communication and exchange of information are done locally between users' information apparatus and the selected output device, all within the range of the users watchful, and inspecting eyes.

10016530 . 110 1101

[0016] Accordingly, the present invention provides a convenient method of digital printing in which a user need not pre-install a device-dependent printer driver on an information apparatus in order to print.

[0017] This invention can provide an easy, friendly and convenient process for printing. Unlike conventional printing, a user need not manually install a printer driver from a CD, floppy disk, or download it somewhere from an existing static network. With the apparatus and process of present invention, all the software components, data, or printer drivers can be loaded to the information apparatus through easy and automatic synchronization process between the information apparatus and a printer having an output controller of present invention. As a result, a mobile user with an information apparatus equipped with an output manager of present invention can print pervasively virtually at any time, anywhere, to any printer.

[0018] Furthermore, this invention can provide a convenient method allowing users to print to any printer with or without connection to a static permanent network for both the information apparatus and the output device. Through local communication and synchronization directly between the information apparatus and the output device, hardware and software installation for static network connectivity may not be necessary. Therefore printing costs may be reduced.

[0019] In addition, this invention can provide a more reliable printing method than conventional remote network printing methods or processes. By communicating locally between an information apparatus and an output device, the printing process does not depend on the perfect function of each element of a network chain. This chain may include, for example, network connections, network servers, application servers, service providers and application providers. Therefore, the chance of a successful printing under present invention is greatly enhanced as processing and communication involves fewer elements.

**[0020]** This invention can further provide a more secure printing process than conventional network printing processes and methods. Uploading or downloading data or software components such as a printer driver from an output device immediately in front of user provides a more secure feeling than dealing with components from a remote location in the network. Components stored in a large network may be perceived to have more exposure to tampering (e.g., contaminated with a virus). An erroneous module or component in the network caused by a human or software or hardware error may also disrupt the function of the information apparatus. Furthermore, sending a confidential digital document or print data directly to the printer in front of user provides a more secure feeling than if the document is routed through a large network with a plurality of users and servers. There are greater possibilities of the document being intercepted, viewed, copied, stored or finally printed on a wrong printer in conventional network printing.

**[0021]** In accordance with present invention, an electronic system and method of pervasive output allow an information apparatus to output digital content conveniently to virtually any output device. The information apparatus may be equipped with a central processing unit, input/output control unit, storage unit, memory unit, and wired or wireless communication unit or adapters. The information apparatus preferably also includes output manager that may be implemented as application software, a client application, or device driver (printer driver in case of printer). The output manager may include management and control capabilities with hardware and software components including for example one or more communication chipsets residing in its host information apparatus.

**[0022]** The output manager in the information apparatus may be capable of communicating with, managing and synchronizing data or software components with an output device equipped with an output controller of present invention. Examples of output devices include, without

10015630-10101



limitation, printers, fax machines, copiers, image or video display devices, monitors, display screens, projectors, and audio output devices.

**[0023]** The output controller may be a circuit board, card or software components residing in an output device. Alternatively, the output controller may be connected externally to an output device as an external component or "box." The output controller may be implemented with one or a combination of embedded processor, software, firmware, ASIC, DSP, FPGA, system on a chip, special chipsets, among others. In another embodiment, the functionality of the output controller may be provided by application software running on a PC, workstation or server connected externally to an output device.

**[0024]** The output controller may include a processing unit, memory/storage unit and communication adapter unit, among others. The storage or memory unit of the output controller may store device drivers, software components or objects encapsulating device dependent data, algorithms, and code. Device dependent data may be, for example, parameters and information about the output device and output controller. Software components and objects may encapsulate for example code or executables of algorithms necessary for converting or encoding an image or document description or language into print data or output data compatible with a specific output device. The output manager may manage the process of uploading data and or software components from an output controller to an information apparatus in order to, for example, enable the information apparatus to generate and transmit print data or output data accurately to the output device or the output controller.

**[0025]** Output data (or print data in the case where the output device is a printer) may be specific data, instructions, page description language, markup language, graphics and or image file format among others. Print data may also be encoded or compressed with one or more compression or encoding techniques. Furthermore, print data may use one or more

10015530-110101

open standards or one or more proprietary techniques and formats, or a combination. Output data (or print data for printers) refers to the electronic data sent from an information apparatus to the output device.

**[0026]** One implementation of a digital content output process of this invention includes the following steps.

1. A user requests to print a digital content from his/her information apparatus.
2. The information apparatus detects available output devices by a discovery process.
3. The information apparatus exchanges service information with output controllers associated with the available output devices in a service negotiation process. The user may then select one or more output devices based on the service information provided.
4. The information apparatus communicates with output controller to identify and upload the necessary components to enable output to a specific output device as part of a synchronization process. The uploaded components or data may then be installed or configured in the information apparatus.
5. The digital content intended for output is processed by one or more components to be compatible with the output device. The one or more components include the information apparatus with an output manager, an output controller and the output device.
6. The printer engine or display engine or audio engine generates final output.

**[0027]** In the output process of present invention, an output manager residing in the information apparatus may participate in, coordinate and manage the communication and exchange of information/components between its host information apparatus and output controllers.

#### **Brief Description of the Drawings**

**[0028]** Fig. 1 is a block diagram of an output system that can implement the process and apparatus of the present invention.

1001630-10101

- [0029] Figs. 2A and 2B are block diagrams illustrating exemplary configurations of hardware components of wireless communication units.
- [0030] Fig. 3A-3C illustrates various configurations and implementations of output manager with respect to an information apparatus.
- [0031] Figs. 4A-4F illustrate various configurations and implementations of output controller with respect to a printer or output device.
- [0032] Fig.5 is a flow diagram of a pervasive output process of the present invention.
- [0033] Fig. 6 is a flow diagram of an exemplary implementation of service negotiation process optionally included in the output process of Fig. 5.
- [0034] Fig. 7 is a flow diagram of different possible configuration of print job process included in the output process of Fig. 5.
- [0035] Figs. 8A-8G show a series of graphical user interfaces (GUIs) rendered at different times on a display screen of an information apparatus.
- [0036] Fig. 9A-9B is a block diagram of a conventional printing system or printer.
- [0037]

#### **Detailed Description of Preferred Embodiments**

- [0038] FIG 1 is a block diagram illustrating exemplary components of an electronic communication system that may provide an operating environment for the present invention and includes an information apparatus 100, which is a computing device with processing capability. As illustrated in Fig. 1, examples of information apparatus 100 may include, but are not limited to, a desktop personal computer (PC), laptop computer, palmtop computer, handheld or mobile device, personal digital assistant (PDA), smart phone, screen phone, set-top box, e-book, Internet pad, digital camera, Internet appliance, pager, etc.
- [0039] Information apparatus 100 may contain components (not shown) such as a processing unit, a memory unit, a storage unit and an input/output control unit, as are known in the art. Information apparatus

100 may also contain an interface (not shown) for interactions with users. The interface may be implemented in software or hardware or a combination. Examples of such an interface include, without limitation, a mouse, a keyboard, a touch-sensitive or non-touch-sensitive screen, one or more push buttons, soft keys, a stylus, a speaker, a microphone, etc.

**[0040]** An information apparatus 100 may be a dedicated single task device (e.g. email terminal, web terminal, e-book, etc) or a general-purpose computing device with multiple features and functions. These multiple functions and features may be implemented by one or more software applications (e.g., applications 302, Fig. 3) installed in the information apparatus 100, which may or may not include an operating system. A digital camera 112 equipped with processing capability and feature sets of the present invention is also an example of an information apparatus 100.

**[0041]** Some information apparatuses 100 (e.g., dedicated devices) may be pre-configured by manufacturers with fixed functionalities and features. Other information apparatuses 100 may allow users to install additional hardware components and application software (e.g., applications 302, Fig. 3) to expand or change functionality. Examples of functionalities and applications of information apparatuses 100 may include, for example, e-mail, messaging, voice communication, web browsing, image acquisition, text processing, graphics applications, document or image editing, and output functionalities, among others.

**[0042]** Functionalities and feature sets of an information apparatus 100 may be implemented in software or hardware or a combination of both. When features are implemented in software, this software may be installed by the manufacturer or by users. Application software may be implemented using embedded software running on embedded processors or it may run on a specific operating system. Some or all or combinations of applications and feature sets may also be implemented in hardware or in silicon. Some functionality or feature sets may be

FOOTNOTES

implemented in special chip sets and may include one or more or combinations of, for example, application-specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), digital signal processors (DSPs), systems on a chip, firmwares, etc.

**[0043]** Information apparatus 100 may or may not contain an operating system. In an implementation having an operating system, the operating system may provide a variety of application programming interfaces (APIs) or object models (e.g., APIs 300, Fig. 3) for software applications to interface and communicate with device drivers. Device drivers in an information apparatus 100 control and communicate with peripheral devices such as printers, copiers, fax machines, display monitors, screens, etc. Several different applications can share a common device driver through the use of APIs or interfaces or object models provided by an operating system, as described below with reference to Fig. 3A. In implementations having no operating system or an operating system that does not support or provide a device driver API, each application may have to implement its own device driver or its own output and communication capability, as described below with reference to Figs. 3B and 3C. For example, to output a digital document to a printer, each application may have to implement its own printer-specific driving, processing and/or communication capability.

**[0044]** In one implementation, information apparatus 100 includes an output manager 308 (Figs. 3A-3C), which is typically implemented in software. Output manager 308 may be implemented or included in a client application, or as a stand-alone application, or as part of another application software 302B (Fig. 3B), or in the form of a device driver (e.g., a printer driver, in case of a printer being the output device), which may be invoked, shared and used by one or more application software programs 302 (Figs. 3A-3C).

**[0045]** One task for output manager 308 is to manage and coordinate communication between information apparatus 100 and one or more

1001630-10301

output controllers 120. Output manager 308 may, for example, implement an open-standard or a proprietary communication protocol to directly or indirectly interact with, manage, or utilize functionalities provided by hardware components residing in a host information apparatus 100. Such hardware components may include, for example, a processing unit, a memory unit and communication chipsets (not shown). Output manager 308 may include one or more of the following features and functionalities:

- Output manager 308 may communicate and interact with one or more software applications 302 included in a host information apparatus 100 to provide pervasive output capability or feature to those software applications 302. The type of information passed from the software applications 302 to the output manager 308 may be, for example, a digital document or digital content data that needs output. In some instances, application software 302 may invoke or launch the output manager 308 as a separate application (as shown in Fig. 3C) or as an integrated part of the application software (as shown in Fig. 3B). Examples of possible communication and management between output manager 308 and application software 302 may include, among others, messages, objects models, instances, procedural calls, program invocation and APIs, which may or may not be provided by an operating system.
- Output manager 308 may manage and maintain wired or wireless communications with output controller 120. Output manager 308 may interact with functionalities implemented by hardware components of its host information apparatus 100. As an example, the output manager 308 may manage and interact with the communication unit in the information apparatus 100. Figs. 2A and 2B show an example of communication units using wireless adapters, as described below in greater detail. A user may interact with the output manager 308

through a user interface provided by output manager 308 through the host information apparatus 100.

- Output manager 308 may help discover available output devices 140, verify communication connections, identify devices available or compatible for service, or send or broadcast service requests to one or more devices.
- Output manager 308 may communicate with output controller 120 to identify necessary software components, data, or any other device dependent information or parameters, if any, that need to be uploaded from the output controller 120 to the host information apparatus 100 to enable pervasive output. Software components may be, for example, a device driver, an application, a special code or algorithm, an executable object or device dependent data, parameter, information, etc.
- Output manager 308 may coordinate with output controller 120 to upload the necessary components or data from the output controller 120 to its host information apparatus 100 so that the user can output digital content pervasively to that output controller 120 or its associated output device 140.
- Output manager 308 may perform raster image processing operations (e.g. rasterization, color management, half-toning operations) or convert digital content from its original format into print data compatible for input to a specific output controller 120 or output device 140. The digital content may be created, stored or received by the information apparatus 100. For example, an output manager 308 may convert digital document information from a certain format or language into a specific print data or intermediate format or language such as PostScript, PCL, EMF, XML, TIFF, JPEG, RTL, among others, for output.

10016330.100101

- Output manager 308 may coordinate or manage the communication between a host information apparatus 100 and output controller 120 to send or transmit the print data to the output controller 120.

**[0046]** Output controller 120

**[0047]** An output controller 120 may be implemented as software or hardware and may be connected to or integrated within an output device 140. Output controllers 120H illustrate hardware implementations in the form of a circuit board or card that is installed or integrated into an output device 140, such as a printer, copiers, fax machine, display screen among others. In another configuration (not shown), output controller 120 may be implemented as a software application or component that is installed or integrated into an output device 140. Output controller 120G illustrates a hardware implementation in the form of an external component or "box" that is separately connected to one or more output devices 140.

**[0048]** Whether externally connected to or internally installed in output device 140, output controller 120 may be implemented with hardware, software, or both, that perform the feature sets and functionalities of the output controller 120 described herein. In the case of output device 140 being a printer, output controller 120 can also sometimes be called a printer server. Exemplary configurations and implementations of output controller 120 are described below with reference to Fig. 4.

**[0049]** As one example, output controller 120 may be implemented as a computer connected to one or more output devices 140. For example, a conventional PC, workstation or server may serve the purpose of an output controller 120. In this case, the functionalities of output controller 120 may be implemented using application software installed in the computer (PC, server, or workstation), with the computer connected to the output device 140.

**[0050]** An output controller 120 installed or connected to an output device 140, such as a printer, enables the output device 140 to communicate



and negotiate services with information apparatus 100. In one implementation, an information apparatus 100 may communicate with an output controller 120 connected to a printer or an output controller 120 internally installed within a printer to obtain device dependent data or software components such as device specific printer driver. An application such as an output manager 308 residing in the information apparatus 100 may install and configure the uploaded components or data, allowing information apparatus 100 to print to that printer directly. An example of an output process is described below with reference to Fig. 5.

**[0051]** Functionalities and feature sets of output controller 120 may include one or more of the following:

- Provide service to clients – Output controller 120 may actively search for available clients, including information apparatus 100 or output manager 308 or other communication units, to which clients' output service could be provided. Alternatively or in combination, an output controller 120 may passively wait to be discovered by one or more clients, including information apparatus 100, to provide output service to the client or clients.
- Negotiation of services with information apparatus 100 - Output controller 120 may communicate and exchange information with information apparatus 100 as part of negotiating the output services to be provided. As an example, in communication with an information apparatus 100, an output controller 120 may inform the information apparatus 100 of the make, model, identification, version, type of input language, type of device driver software, type of services provided, type of components available for upload, etc. for a selected output device 140 such as a printer. As another example, an information apparatus 100 may send one or more messages to an output controller 120 inquiring about what software component or data, if any, the information apparatus 100 needs to upload from the output controller

1004630-10404

120 to enable output to a specific output device 140.

- Provide security functionalities such as authentication and encryption – Authentication is used to prevent unwanted access to services, while encryption is used to prevent eavesdropping. Security procedures may be implemented by software, hardware or a combination of both, in various steps and stages of communication between the output controller 120 or the output device 140 and the information apparatus 100.
- Payment or transaction management and services – A user may be charged a service fee when, for example, providing output to a printer 140 or other output device 140. Output controller 120 may have the ability to calculate and process payment according to the services requested by or rendered to a client. Output controller 120 may process payments in various forms such as cash, E-cash, credit card, debit card, E-card, smart card, virtual cash, etc.
- Storage capability – Output controller 120 may include a memory or storage unit and may store within it one or more of the following: application software, device or printer driver, software components, device dependent data or parameters, user interfaces, etc. Software components or data stored in the output device 140 or the output controller 120 may be uploaded to the information apparatus 100 in order for the information apparatus 100 or the output manager 308 to drive the output device 140. Software components may be executable or partially executable programs or data.
- Synchronize with information apparatus 100 – After output controller 120 or the output manager 308 in information apparatus 100 identifies the components (software component, data, information or parameters) necessary to enable output or printing, the output controller 120 may coordinate with the output manager 308 to upload to the information apparatus 100 the components stored in a memory or storage unit of output controller 120.

1001E90-110101

- Provide management tools – Multiple information apparatuses 100 may request service from the same output device 140 at the same time or at overlapping times. Management tools such as queuing and spooling of print jobs, quality of service, etc. may be provided by the output controller 120.

**[0052]** As indicated above, output controller 120 may be implemented in a variety of ways and may include hardware, software, or a combination of the two. In hardware implementations, for example, output controller 120 may include components using one or more of hardware types such as ASIC, DSP, FPGA, firmware, system on a chip, and communication chip set. Output controller 120 may also include embedded processors with software components or embedded application software to implement its feature sets, and functionalities. An output controller 120 may or may not contain an operating system.

**[0053]** Output controller 120 typically includes a memory or storage unit (ROM, RAM, flash memory, disk drive – not shown), which may store one or more software applications, software components and data. Some software applications may run in the output controller 120 to perform its functionalities (e.g., communicate with information apparatus 100).

**[0054]** The memory or storage unit of an output controller 120 may also store data and one or more software components that may be available to be uploaded to or synchronized with an information apparatus 100 during communication with output manager 308. A software component may be a device driver or complete application software that can run on the information apparatus 100. Once such a software component is uploaded and installed, the information apparatus 100 can drive or output to that specific output device 140.

**[0055]** A software component may also be a partial device driver or partial application software or partial software component. A software component may be an executable or partially executable program, object

or code. Software components may encapsulate information, algorithms, and data. Examples (of algorithms that may be included in the software components include compression/decompression methods, file conversion, graphics representation and rendering, image processing and enhancement operations, color space transformation and data conversion, encoding/decoding techniques, color matching and management methods, image transformation, interpolation methods halftoning techniques, user interface software codes etc. Examples of data that may be included in software components include, for example, a color table, color space information, halftoning tables, output device type, output resolution, bit depth, input type, format or language, communication protocol type, output status, device identification and signature, graphics, image and text descriptions etc.

**[0056]** Such software components may be uploaded to output manager 308 to facilitate communication and generation of print data or output data suitable, compatible or optimized for the output controller 120 or the output device 140. The output controller 120 and output manager 308 manage and negotiate the proper components to be uploaded to the information apparatus 100.

**[0057]** Not all situations require uploading an entire or complete driver or application to output manager 308 or information apparatus 100. In some implementations uploading only a software component may be sufficient. For example, there are many inkjet printers that commonly use one-bit or multi-bit CMYK printing technology, and sometimes have similar resolutions. An information apparatus 100 may pre-install a more generic printer driver, where this printer driver contains the device independent portion of the software code, algorithms, and data that is common or can be used for this entire class of printers. Device-independent code may include, for example, a rasterization process (including half-toning and color management methods) and image enhancement of the digital document that is suitable for use with this

10016630-110101

entire class of printers. In this example, it is feasible to upload only a smaller software component or data that includes the device-dependent components for a particular printer or output device 140. The device-dependent portion or component or data may include, but is not limited to, color tables, halftoning tables, output resolution, bit depth, compression/decompression methods, color space conversion, encoding algorithms or conversion software that is specific to that printer or output device 140.

**[0058]** Another example of a software component stored in the memory unit of an output controller 120 for uploading to an information apparatus 100 is an installation wizard or a user interface to capture a user's preferences for printing or output operation. Examples of user preferences in the case of printing may include, without limitation, color and print quality parameter adjustments, number of copies, number of cards per page, number of pages per sheet, duplex, portrait or landscape, security information, etc.

**[0059]** Once the software components or data are uploaded or installed in information apparatus 100, the information apparatus 100 can drive or output to the output device 140. For example, an application 302 in the information apparatus 100 can print a digital document or image to a printer.

**[0060]** Different information apparatuses 100 may require different software components to be uploaded in order to drive a specific output device 140. Accordingly, a variety of software components and data or device drivers for a plurality of information apparatuses 100, operating systems, and platform types may be stored in the memory or storage components of an output controller 120.

**[0061]** Output controller 120 may contain an embedded operating system to manage its applications. With an operating system, some or all functionalities and feature sets of the output controller 120 may be implemented as application software. Additional application software

10016630 110101

may be installed or upgraded to newer versions in order to, for example, provide additional functionalities or bug fixes.

**[0062]** Output controller 120 may contain a user interface that allows a user to configure and manage the functions provided by the output controller 120. The user interface of the output controller 120 maybe accomplished by, for example, using display screens, soft keys, pushbuttons, mouse, stylus and keypads to allow users to interact with and manage the functions provided by the output controller 120. Examples of such functions may include any or all of an on-off power switch, status display or indicator, cancel current job, or reorder existing jobs, etc.

**[0063]** Output controller 120 typically also includes a communication or adapter unit, which allows output controller 120 to communicate with other devices including information apparatus 100 through wired or wireless connections. Exemplary configurations of wireless adapter or communication unit are described below with reference to Fig. 2.

**[0064]** When output controller 120 is implemented as firmware, or an embedded application, the configuration and management of the functionalities of output controller 120 may be optionally accomplished by, for example, using controller management software in a host computer (not shown). A host computer may be a desktop personal computer (PC), workstation, or server. The host computer may be connected locally or through a network to the output device 140 or controller 120. Communication between the host computer and output controller 120 can be accomplished through wired or wireless communication. The management application software in the host computer can manage the settings, configurations, and feature sets of the output controller 120. Furthermore, a host computer configuration application may download or install application software, software components or data in the output controller 120 for the purpose of

10016530-10401

upgrading, updating, and or modifying the features and capabilities of the output controller 120.

**[0065]** Output device 140 in one implementation includes or is connected to output controller 120 described above. Therefore, functionalities and feature sets provided by output controller 120 are automatically included as part of the functionalities of output device 140. The output device 140 may, however, implement or include other controllers or applications that provide at least partially the features and functionalities of the output controller 120.

**[0066]** Therefore, the output device 140 may include some or all of the functionalities and feature sets provided by the output controller 120.

**[0067]** Output device 140

**[0068]** Output device 140 is an electronic system capable of outputting digital content regardless of whether the output medium is a substrate (e.g., paper), a display image, a projection, or sound. A typical example of output device 140 may be a printer, which outputs digital documents containing text, graphics, images or any combination onto a substrate. Output device 140 may also be a display device capable of displaying still images or video, such as, without limitation, televisions, monitors, and projectors. Output device 140 can also be a device capable of outputting sound. Any device capable of playing or reading digital content in audio (e.g., music) or data (e.g., text or document) formats is also a possible output device 140. A printer (including a fax machine, copier, etc.) is frequently referred to herein as the exemplary output device 140. However, it should be recognized that the present invention applies as well to output devices 140 other than printers.

**[0069]** Fig. 9A is a block diagram of a conventional printing system or printer 900A having three basic components: raster image processor (RIP) 902, a memory buffer 904, and a marking engine 906. Raster image processor 902 converts digital content into a bitmap suitable for

FOOTNOTES

printing; memory buffer 904 holds the rasterized image ready for printing; and marking engine 906 transfers colorant to a substrate (e.g., paper).

**[0070]** Marking engine 906 may use any of a variety of different technologies to transform a rasterized image to paper or other media or, in other words, to transfer colorant to a substrate. The different marking or printing technologies that may be used include both impact and non-impact printing. Examples of impact printing may include dot matrix, Teletype, daisywheel, etc. Non-impact printing technologies may include inkjet, laser, electrostatic, thermal, dye sublimation, etc.

**[0071]** Marking engine 906 and memory buffer 904 of a printer 900 form its printer engine 908, which may also include additional circuitry and components, such as firmware, software and or chips or chipsets for decoding and signal conversion etc. (not shown). Input to a printer engine 908 is usually a final rasterized print data generated by the raster image processor 902. Such input is usually device dependent and printer specific. The printer engine 908 may take this device dependent input and generate output pages.

**[0072]** Raster image processor (RIP) 902 may be located within the printing device itself (as shown in fig 10A) or externally implemented as hardware, software, or a combination. As an example, RIP may be implemented in a software application or device driver in the information apparatus. A RIP may also reside within a printer controller, a print server or an output controller of present invention. Examples of raster image processing operations may include image and graphics interpretation, rasterization, segmentation, color space transformation, image enhancement, color correction, halftoning etc.

**[0073]** When a RIP 902 is located inside an output device 140, the RIP 902 is usually included in a printer controller 910 that may, for example, interpret, convert and/or rasterize input print data into a final format, language or instructions that printer engine 908 can understand. A laser printer with a faster printing speed is an example that falls in this

FIG. 10A



category. For example, a PostScript printer controller 910 may contain a postscript interpreter (not shown) and a raster image processor 902 (RIP). Postscript interpreter interpret the postscript language and the raster image processor 902 may process the digital document and converting it to a printer specific format, language or instructions that can be accepted by the printer engine 908. In this example, a device-dependent printer driver (e.g., a Postscript driver, not shown) may be needed in the information apparatus 100 to convert digital document from its original format into print data (e.g. in PostScript) that is acceptable to the printer controller 910.

**[0074]** Another example of page description language is PCL (Printer Command Language) from HP (Hewlett Packard). In the same way as postscript, a printer equipped with PCL printer controller 910 can take PCL as input. Information apparatus 100 usually needs to have PCL printer driver in order to drive or print to a printer equipped with a PCL controller.

**[0075]** Although the illustrations above use PostScript and PCL as examples, a variety of other page description languages, markup languages, image formats, graphics formats, and file formats may also be used as input print data 920 to a printer 900 or output device 140. Examples of possible inputs other than PostScript and PCL may include without limitation, EMF, XML, HTML, among many others. Some printer manufacturers may also use a combination of proprietary or non-proprietary page description languages, markup languages, file formats, graphics and image formats, color spaces, metafiles, encoding, decoding, compression or decompression etc. for the print data 920. Print data 920 sent to printers with printer controllers 910 are usually intermediate descriptions of a digital document that may require further interpretation, processing and/or conversion before it can be send a printer engine for output. A printer controller 910 may interpret and process the input intermediate print data information into the final format

that can be understood by the printer engine 908. Regardless of the type of print data 920, users usually have to pre-install a device-specific driver in their information apparatus 100 in order to output the proper language, format, or file that can be accepted by a specific printer 140.

**[0076]** Some output devices 900B do not have a printer controller 910 as shown in Fig. 9B. A typical example is a lower-cost inkjet printer. RIP in this example may be implemented in a software application 302 or device driver or output manager 308 of present invention in the information apparatus 100. Therefore, the rasterization process may occur in the information apparatus 100 that outputs to such a printer. A device-specific driver or application may need to reside in the information apparatus 100 to rasterize and convert the digital document from its original format into final print data 920B (e.g., compressed CMKY data with one or more bits per pixel) that can be understood by a particular printer engine 908B.

**[0077]** In present invention, for the printer 900B, RIP can be integrated within the output manager 308 or application software 302 in the information apparatus 100. Alternatively, RIP may also be integrated as a part or a function of any one of output controllers as shown in configurations 120D, 120E, and 120F.

**[0078]** The output controller 120 may be connected externally to an output device 140 or integrated internally into the output device 140. If an output device 140 such as a printer already includes a printer controller 910, the output controller 120 may be implemented serially or cascaded with the printer controller 910 as two separate controllers. Alternatively, output controller 120 can be integrated with a printer controller to become a "combined controller" 120C. A combined controller (e.g., 120C or 120F) has functionalities of both printer controller 910 (e.g., input interpretation and or raster image processing) and output controller 120 of the present invention. Other configurations and implementations of output controller 120 and printer controller 910 are also possible. The

1001630 110101

various possible configurations and implementations of output controller are described below with reference to Fig. 4.

**[0079]** Communication unit

**[0080]** Communication links 150 and 160 between information apparatus 100 with an output controller 120, whether externally connected and or internally integrated, may be variously implemented. In one implementation, information apparatus 100 communicates with output controller 120 through wireless connections such as infrared or radio links. Examples of wireless connections technology include without limitation IrDA, home RF, Bluetooth, IEEE 802.11, HiperLan2, among others. However, wired connections such as serial interfaces, parallel interfaces, USB interfaces, Fire Wire (IEEE 1394), Ethernet and token ring network among others may also be implemented in the present invention. Adapter pairs may be incorporated into each communicating node (e.g., information apparatus 100 and output controller 120) to conduct communications by performing signal conversions.

**[0081]** In the case that output controller 120 is installed as an external component or "box" (as shown in Fig. 4A), the communication link 170 between output controller 120 and output device 140 may also be implemented using wired or wireless adapters.

**[0082]** Figs. 2A and 2B are block diagrams illustrating two exemplary configurations and implementations of wireless communication adapters that may be included in an information apparatus 100, output controller 120 or output device 140. Referring to Fig. 2A, a radio adapter 200 may be implemented to enable data/voice transmission among devices through radio links. A RF transceiver 214 coupled with antenna 216 is used to receive and transmit radio frequency signals. The RF transceiver 214 also converts radio signals into and from electronic signals. The RF transceiver 214 is connected to a RF link controller 210 by an interface 212. The interface 212 may perform functions such as

analog-to-digital conversion, digital-to-analog conversion, modulation/demodulation and other data conversion functions.

**[0083]** RF link controller 210 implements real-time lower layer (e.g., physical layer) protocol processing that enables the hosts (e.g., information apparatus 100, output controller 120, output device 140, etc.) to communicate over a radio link. Functions performed by the link controller 210 may include, without limitation, error detection/correction, power control, data packet processing, data encryption/decryption and other data processing functions.

**[0084]** A variety of radio links may be utilized, including a group of competing technologies operating in the 2.4 GHz unlicensed frequency band. This group currently includes Bluetooth, Home radio frequency (Home RF) and implementations based on the IEEE 802.11 standard. Each of these technologies has a different set of protocols and they all provide solutions for wireless local area networks (LANs). Interference among these technologies could limit deployment of these protocols simultaneously. It is anticipated that new local area wireless technologies may emerge or that the existing ones may converge. Nevertheless, all theses existing and future wireless technologies may be implemented in the present invention to transmit data between different devices without limitation and therefore in no way depart from the scope of present invention.

**[0085]** Among the current available wireless technologies, Bluetooth requires relatively lower power consumption. Bluetooth has its own protocol stack and is designed for short range (10 meters), point-to-multipoint voice and data transfer. It is based on a frequency-hopping version of spread spectrum. Seventy-nine hop frequencies are utilized beginning at the lowest frequency of 2402 MHz and each of the 79 hop frequencies is 1 MHz above the next lower frequency. Bluetooth-enabled devices operate in piconets, in which several devices, using the same hopping pattern or sequence, are connected in a point-to-

10016630-100101

multipoint system (piconet). One device (master) in each piconet determines how the bandwidth is allocated to other devices (slaves). As many as 10 piconets of 8 devices each can operate simultaneously.

**[0086]** Referring to Fig 2B, infrared (IR) adapters 220 may be implemented to enable data transmission among devices through infrared transmission. The IR adapters 220 may be conveniently implemented in accordance with the Infrared Data Association (IrDA) standards and specifications. In general, the IrDA standard is used to provide wireless connectivity technologies for devices that would normally use cables for connection. The IrDA standard is a point-to-point (vs. point-to-multipoint as in Bluetooth), narrow angle, ad-hoc data transmission standard designed to operate over a distance of 0 to 1 meter and at speeds up to 4 Mbps.

**[0087]** Configuration of infrared adapters 220 may vary depending on the intended rate of data transfer. Fig 2B illustrates one embodiment of infrared adapter 220. Transceiver 226 receives/emits IR signals and converts IR signals to/from electrical signals. A UART (universal asynchronous receiver/transmitter) 222 performs the function of serialization/deserialization, converting serial data stream to/from data bytes. The UART 222 is connected to the IR transceiver 226 by encoder/decoder (ENDEC) 224. This configuration is generally suitable for transferring data at relatively low rate, for example 115.2 kbps or below. Other components (e.g. packet framer, phase-locked loop) may be needed for higher data transfer rates.

**[0088]** Figs 2A and 2B illustrate exemplary configurations of wireless adapters. Such adapters are implemented into devices (e.g., information apparatus 100, output controller 120, output device 140, etc.) to enable wireless communications among/between them. Wired links, however, such as parallel interface, USB, firewire interface, Ethernet and token ring networks may also be implemented in the present invention by using appropriate adapters and configurations.

**[0089]** Different Output Manager Configurations and Processes

**[0090]** Figs. 3A – 3C illustrate different implementations of output manager 308. Fig 3A shows an exemplary configuration of output manager 308 in which its host information apparatus 100 includes an operating system. Such information apparatuses 100 may include, without limitation, desktop PC, laptop PC and handheld or palmtop computer, among others. The operating system provides API's or object models 300 for software applications 302 to interface and communicate with various device management applications sometimes called device drivers. Output manager 308 in this case may be characterized as one of the device management applications or device drivers. Different software applications 302A can share the same output manager 308 through the use of API's 300 provided by an operating system.

**[0091]** As shown in Fig 3A, email application 302 A1, document creation application 302 A2 and Internet browsing application 302 A3 (referred to generally or together as application software 302) may operate on top of or through an operating system included in the information apparatus 100. The operating system provides or supports API's or object models 300 for these software applications 302 to communicate with and access utilities and services provided by output manager 308. Application software 302 and output manager 308 may communicate with each other through a variety of means using for example file transfer, metafile, messages, instances, procedural calls, etc., which may or may not be provided by the operating system or its API's or object models 300. The output manager 308 communicates and synchronizes with output controller 120 to enable pervasive output. Therefore, a user may output digital contents from any of these software applications 302 (email, document creation, Internet browsing, etc.) to an output device 140 equipped with or connected to output controller 120.

**[0092]** It should be recognized that the three software applications 302(1) – 302(3) illustrated in Fig. 3 are merely examples of applications that

1001630-10101

may be included in the information apparatus 100. Various other software applications 302 (e.g., messaging, digital imaging, graphics, video, audio, etc.) may also exist in the same information apparatus 100 and they may all access the utilities provided by the output manager 308. It should also be recognized that other device management applications or drivers might also be included in the information apparatus 100. These device management applications or drivers may be implemented in the same way as output manager 308 so that they can be accessed by other software applications supported by the operating system.

**[0093]** Figs. 3B and 3C illustrate exemplary configurations of output manager 308 when information apparatus 100 includes, respectively, no operating system or a less capable operating system that does not provide API's or object models 300 for software application 302 interactions. Examples of such an information apparatus 100 may include, without limitation, PDA, smart phone, pager, Internet pad, email terminal, digital and video cameras, ebook and other dedicated devices etc. In this case, each software application 302 in the information apparatus 100 provides or implements the capabilities to enable pervasive output. As shown in Fig 3B, each software application 302 B1 – 302 B3 integrates output manager 308 or includes the features of output manager 308 to enable pervasive output. It should be recognized that, even in an environment where there is a complete operating system, application software 302 can still decide to implement its own output or device management capabilities (as shown in Fig 3B) and not depend or use the functionalities or API's or object models 300 provided by an operating system to facilitate access of output manager as a separate application.

**[0094]** Instead of integrating the output manager 308 into application software 302, an alternative implementation is shown in Fig 3C. The software applications 302 C1 – 302 C3 may establish direct links with the

output manager 308, which may be used as a helper program that can be invoked or launched by each software application 302 as needed.

**[0095]** The communication and exchange of information between output manager 308 and other software applications 302 can be accomplished by one or more of many available techniques, such as passing, transferring or exchanging objects, messages, procedural calls, files, metafiles, etc. Some of these techniques may be facilitated by a set of APIs or object models provided by the application software 302 or output manager 308 or the operating system. The exact technique may depend for example on the type of application, platform, language, and developer preference etc. This may be easily implemented and recognized by an average software engineer skilled in the art.

**[0096]** The above are merely exemplary implementations of output manager 308. Due to the diversity of feature sets and capabilities of operating systems, other implementations of output manager 308 are also possible. As an example, some operating systems may provide partial APIs or object models to support limited output capability. In this case, software applications 302 may rely on the limited output capacity to interact with device management applications, or the software applications 302 can implement or incorporate partially other desired output features of the output manager 308 using the configurations or combination of configuration illustrated in Figs 3B or 3C.

**[0097]** Different Output Controller Configurations and Processes

**[0098]** Figs. 4A – 4F illustrate various alternative configurations and implementations of output controller 120. As described above, a printer 900A (Fig. 9) may include within it a RIP 902 or printer controller 910. Such printer 900A is usually more powerful and therefore more expensive than printers without these features. One example of such printer 900A is high-speed laser printer. Output controller 120 may be variously implemented in printers regardless of whether they include a printer controller.



[00102] Some printers do not include a raster image processor or printer controller 910, as illustrated in Fig. 9B, and as a result are usually less expensive. An example of this type of printer is a low-cost desktop inkjet printer. This type of printer may use a proprietary or published input format. For example, the input to an inkjet printer may consist of specific proprietary compressed CMYK data with one or more bits per pixel input. Application software 302 or printer drivers included in information apparatus 100 usually perform raster image processing. It is therefore

necessary that the proper device driver or application software runs the raster image processing on an information apparatus 100 before printing. Accordingly, output controller 120 may be implemented into a variety of printing systems, including printers without capable or complete printer controllers for performing raster image processing functions.

**[00103]** Fig. 4D shows as a fourth implementation of output controller 120D that is installed in an external component or "box," which is connected to a printer 140D that does not include a printer controller or raster image processor. In this configuration, an application software 302 or device driver 308 in information apparatus 100 would typically perform the rasterization function. However, some or all operations of raster image processing may also be implemented in the output controller 120D. Any print data sent to the printer 140D from the output controller 120D is preferred to be in a final format that can be understood and processed by the printer engine of the printer 140D. As in other examples, connection between output controller 120D and output device 140D can be wired or wireless. Connection between information apparatus 100 and output controller 120D is preferred to be wireless, for example.

**[00104]** Fig. 4E shows as a fifth implementation of an output controller 120E that is incorporated within printer 140E as one or more circuit boards or cards and may contain software and applications running on an embedded processor. As with printer 140D (Fig. 4D), printer 140E does not include a printer controller or raster image processing capability. Accordingly, information apparatus 100 would typically perform rasterization functions in this implementation. However, raster image processing could alternatively be provided completely or partially by output controller 120E, for example, to speed up raster image processing operations for an information apparatus 100 having limited processing and memory capabilities.

**[00105]** Fig. 4F shows a sixth implementation, a combined controller 120F that integrates the functionalities of a printer controller and an output

controller into a single external combined controller component or “box” 120F. The two controller functions may share a common processor as well as a common memory space to run applications of the two types of controllers. Under this configuration, either information apparatus 100 or the combined controller 120F could perform or share raster image processing functionality. One exemplary implementation of the combined controller 120F shown in Fig 4F is to use an external computing device (PC, workstation, or server) running one or more applications that includes the functionality of output controller 120 and printer controller 910.

**[00106]** The above are exemplary implementations of output controller 120, but other implementations are also possible. For example, partial functionalities of output controller 120 may be implemented in an external “box” while the remaining functionalities may reside within an output device 140 as a separate circuit board or integrated with a printer controller. As another example, the functionalities of output controller 120 may be implemented into a plurality of external boxes connected to the same output device 140. As a further example, the same output controller 120 may be connected to and service multiple output devices 140.

**[00107]** Fig 5 illustrates one embodiment of output process using a printer as primary output device 140. It should be recognized, however, the following described process can be similarly applied to other output devices 140 such as fax machines, digital copiers, display screens, monitors, TVs, voice output devices, etc

**[00108]** Step 500 indicates that a user initiates output from an information apparatus 100. For example, a user may want to print or otherwise output an email, receipt, confirmation, map, photo, graphics, web page, one or more slides of a PowerPoint®-brand presentation, or any other document or digital content from an information apparatus 100. The output may be initiated by various means, such as by using a soft key,

10046330-13341

push button, keyboard, keypad, mouse, stylus, software GUI, command or voice activated command, etc. An exemplary user interface with which a user may initiate such a process from an information apparatus 100 is described below with reference to Fig 8. Step 500 and one or more other steps of output process, shown in Fig 5, may be implemented by an output manager 308 residing in a user's information apparatus 100.

**[00109]** Step 502 references a discovery process in which, for example, information apparatus 100 searches for available output devices 140, including display devices, printers, copiers or fax machines among others (collectively refer to as "printers" here for simplicity of discussion), that can perform a requested output or print job. The output manager 308 residing in the information apparatus 100 may participate in, coordinate and manage the discovery process 502. As part of discovery process 502, information apparatus 100 may communicate via a wired or wireless connection with output controllers 120 that are connected to or included in output devices 140.

**[00110]** In some situations, the discovery process 502, or part of it, may be skipped. For example, discovery process 502 may be skipped if a user already knows the output device or printer to which output is to be directed. As another example, a user may set a frequently used printer as a preferred default printer so that discovery process 502 may be partly skipped if the default printer is found to be available. Other examples in which discovery process 502 may be optional or partly skipped involve direct wire line connections or infrared communications. In these cases, a user may directly connect or physically point an information apparatus 100 to a specific output device 140 so that it could be sufficient only to check the connection and establish communication between the devices.

**[00111]** Discovery process 502 may operate in a variety of scenarios. In one scenario, information apparatus 100 broadcasts a service request,

10016630-10101

for example printing, through wired or wireless signals. Service devices, such as output device 140 equipped with output controller 120, “listen to” such signals, “read” the request, “see” whether they can provide requested service, and respond if they can provide the requested service. Alternatively or in combination, all service devices can periodically or continuously announce or advertise the services they provide. Information apparatus 100 “listens to” such announcements and identifies the service it needs. In a third scenario, service devices of the same network (e.g., LAN) register their services with a control point. A control point is a computing system (e.g. a server, controller) that maintains records on all service devices within the same network. Information apparatus 100 may contact the control point to search for the service it needs.

**[00112]** Various protocols may be implemented in the discovery process 502. Wireless communication protocols are preferred. Wired communication, on the other hand, may also be implemented. Examples of applicable protocols may include, without limitation, Bluetooth, HAVi, Jini, Salutation, Service Location Protocol, and Universal Plug-and-play, among others. Other proprietary protocols or combinations may also be implemented in the discovery process.

**[00113]** In the discovery process 502, information apparatus 100 may or may not find an output device 140 available to accept its print or output job. Exemplary factors that may impact the availability of output devices 140 may include, without limitation, the following:

- Physical distance – Some wireless communication links may impose limitations on physical distance between communicating devices. The restrictions on physical distance may vary among different wireless communication technologies. Wired communication, on the other hand, is also subject to the availability and length of cable or other wired lines.

- **Compatibility** – The output process with reference to Fig 5 employs communication and coordination among various devices such as an information apparatus 100 having output manager 308, and output controllers 120 associated with output devices 140. Compatible hardware and software components may need to reside in each of the communicating nodes (e.g., information apparatus 100 and output device 140). Any missing part may cause compatibility problems. Other factors may also cause lack of compatibility. For example, components stored in the memory unit of output controller 120 may not support the operating system or application software 312 included in the information apparatus 100.
- **Security authentication** – Security authentication may be implemented in various steps of communication, including the discovery step 502. Security authentication may be required for example when an output service is restricted to a certain group of users or information apparatuses. When a user requests service, output device 140 or output controller 120 may process an authentication step by automatically detecting the identification or other information provided by the user, output manager 308 or information apparatus 100. As a result of such an authentication process, the information apparatus 100 of a particular user may not be able to discover an available output device 140 because access is not authorized for the user or the apparatus.
- **User specified search criteria** – The user may specify certain search criteria for discovery process 502, such as output device model, quality of service provided, range or price of service charge, etc. Output devices 140 not satisfying user criteria or a default profile may not be shown to the user.

**[00114]** Step 506 indicates that the user is notified if no available output device 140 is found in the discovery process 502. The user may be notified by an application such as the output manager 308 through a user

interface of the information apparatus 100. The information apparatus 100 may also provide more detailed information such as the reasons why the requested service is not available. Examples of possible reasons may include, among others: no output device within limited physical distance, no compatible hardware or software components, technical difficulties, security authentication not satisfied, no subscription to the service, search criteria not met, etc.

**[00115]** Step 508 indicates that alternatives are provided when a requested service is not available at the current moment. The user may be provided with alternatives such as canceling the service request, trying again, or being notified when the requested service is available or the requested output device 140 is identified or becomes available. As an example, the user's information apparatus 100 may not detect any available output devices 140 in the current wired/wireless network. The print requests are then queued or registered within the information apparatus 100. When the user enters a new network having available printers, or when new compatible printers are added to the current network or when a printer becomes available for any reason, the user would be notified of such availability through the information apparatus 100. The user may then output the print job to the newly detected or now available printer with, for example, a mere click of a button. This detection and notification function may be implemented by a software program or as a feature of the output manager 308. This feature may also be implemented in hardware or combination of hardware and software residing in the information apparatus 100.

**[00116]** Step 514 indicates that service negotiation is performed, such as when one or more available printers are discovered in the discovery process 502. In service negotiation process 514, information apparatus 100 exchanges information with output controllers 120 connected to or installed in the available output devices 140. The output manager 308 may participate in, coordinate or manage negotiation process 514 with

FOR OFFICIAL USE ONLY

output controller 120. Information being exchanged may include, for example, the following:

- Type, make and model of each available output device 140, including information about the printer engine, its technology and specifications of the output device.
- Type and parameters of the output device 140 such as font or international character code supported, paper size, resolution, bit depth, color space, number of colors supported, color or grayscale etc.
- Type of service and quality of service provided by each available output device 140.
- Conditions under which the services are provided (e.g., service charge, subscription charge) and whether, for example, transaction and service charges are satisfied.
- Whether the usage or service of each output device 140 is restricted to certain groups, in which case the user may have to provide authentication information to be identified as member of such a group before use of that device is allowed.
- Type, make and model (including operating system) of the information apparatus 100.
- The type of input format, encoding, language, data, color space, and or compression that the output controller 120 or other device may take as input, for example, PostScript, PCL, RTL, XML, etc.
- The type of component or data that may need to be uploaded from output controller 120 to information apparatus 100 so as to enable output to a specific output device.

**[00117]** The information being exchanged may be entered manually by the user or may be automatically detected and or synchronized between the information apparatus 100 and the output controller 120 or it may be partially assisted by the user or partially automated. Some or all of the information being exchanged may be provided to the user through a user interface in the information apparatus 100. The user may monitor the



status or approve certain transactions. One embodiment of the service negotiation process 514 is described below with reference to Fig. 6. Based on the information provided and obtained in the service negotiation process 514, the user may choose one or more output devices 140 that can take the print or output job.

**[00118]** Step 516 indicates that a synchronization process is performed once an output device 140 has been identified and selected. Before synchronization process 516 begins, the information apparatus 100 may communicate with the output controller or controllers 120 of the selected output device or devices 140 to identify what components, if any, need to be uploaded to the information apparatus 100 to enable printing or other output. This communication between information apparatus 100 and output controller or controllers 120 may not be necessary if adequate information has been obtained in the discovery process 502 and or service negotiation process 514. Synchronization process 516 may be optional or it may be partially skipped if the necessary components (e.g., identified in the service negotiation step 514) already exist in information apparatus 100.

**[00119]** After they have been identified, the necessary components or parts of components or data may be uploaded to the information apparatus 100 from output controller 120. Examples of possible components may include, without limitation, one or more of the following: data, device driver, printer driver, application software, software components, metafiles, user interface etc. The output manager 308 may participate in, coordinate or manage the synchronization process 516. Encryption techniques may be implemented in the synchronization process 516 to prevent eavesdropping.

**[00120]** The newly uploaded components may be incorporated into the output manager 308 residing in the user's information apparatus 100. Alternatively or in combination, the newly uploaded components may be installed into the information apparatus 100 as a separate application or

1001630-10101

as part of the output manager 308. The process of uploading and installing may be done automatically or manually by the user, or a combination of the two.

**[00121]** Part or all of synchronization process 516 may be skipped in some instances. For example, the information apparatus 100 may already include all the software components and data or drivers necessary for output to the selected output device or devices 140. This may be due to the components having been uploaded from a previous synchronization process 516 and left undeleted, or alternatively the user may have preinstalled all the necessary components for output in the information apparatus 100. In these cases, no software components are being uploaded and the user may be notified of this status for possible override of synchronization process 516. Alternatively synchronization process 516 may be automatically skipped transparently to the user.

**[00122]** Step 518 indicates that print or output job processing is performed. In this step 518, the print request or print job is processed and digital content or document may be converted to the final format to send to the output device 140 for output. At the beginning of this step 518, the user may be provided with options to select or change or input printing or output preferences such as page layout, number of copies, color/grayscale, print quality parameters, etc. These options may include or offer pre-configured user preferences or factory defaults. Some options, parameters and preferences may be device-dependent while others may be device-independent. For example, device dependent parameters and preferences may be presented after having synchronized or uploaded device dependent components or information from the output controller 120. Device independent parameters and preferences may also be uploaded and synchronized like the device dependent parameters and preferences or may be pre-installed as part of the output manager 308 prior to the synchronization and uploading of information.

10016330-110101

**[00123]** With reference to Figs. 4A-4F, output controller 120 may be implemented in a variety of ways. Different implementations of output controller 120 may impact the procedures described here. The original digital document or content included or generated in an application 312 in information apparatus 100 may be processed by different applications in different locations (application software 312, output manager 308, output controller 120, printer controller) before being converted to a final print data that can be accepted by a printer engine for output. For example, raster image processing of the digital document to convert it to print data suitable to output device 140 may occur entirely or partially in application software 312, output manager 308, output controller 120, or the printer controller, depending on the implementation, configuration, and the type of output system and output device among others. Different processing procedures and embodiments that may be implemented in this step 518 are described in more detail below with reference to Fig. 7.

**[00124]** Step 520 indicates that the printer engine or display engine 908 generates a final output. In this step 520, the processing result of step 518 is sent to a printer engine or display engine 908 of an output device 140 to generate final output on a substrate, for example. Memory buffers may be needed to temporarily store print data before output. Firmware, DSP, ASIC, FPGA, system-on-a-chip, software, or a combination, may be used or involved to convert digital data into signals suitable for the printer engine or display engine 908.

**[00125]** After a print job or output job has been successfully printed or displayed, the user may be provided a choice (not shown) to delete the components that have been uploaded to the information apparatus 100 during the synchronization process 516. This optional feature may be useful when the information apparatus 100 has limited storage space. One implementation of this optional feature is to provide a setting in the output manager 308 in which the user has a choice to (1) always keep the uploaded data or components; or (2) always automatically delete the

10016330.10101

uploaded data or components upon completion of the print job, or upon exit of the application, or upon shutting down of the information apparatus or upon elapse of certain amount of time; or (3) always ask the user whether to delete or keep the uploaded component or data.

**[00126]** The above description illustrates an exemplary embodiment of output process 501. A printer is referred to as output device 140 in the above illustration, however, it should be recognized that output process 501 could be easily applied to other output devices 140 such as fax machines, digital copiers, display screens, TVs, monitors, projectors, voice output devices etc.

**[00127]** A pervasive output process implementing fewer or additional steps may also be possible. For example step 502, the discovery process, may be skipped if the user already knows which printer or printers are to output the content. As another example, a payment-processing step may be added to the above-described output process 501 if the printing service is provided for a fee. Output controller 120, in this case, may provide services such as calculating payment amount or collecting payments in a variety of possible forms, for example, credit card, bank card, cash, E-cash, smart card, among others. The payment information (e.g., credit card number or other credit identification or payment) may be stored in the information apparatus 100 and released to the output controller 120 at the user's choice. Adding additional processes or skipping one or more steps in output process 501 does not, however, depart from the spirit and scope of present invention.

**[00128]** Fig. 6 illustrates an embodiment of service negotiation process 514. In this illustration, it is assumed that available output devices 140 have been identified from the discovery process 502. In the service negotiation process 514, information apparatus 100 exchanges information with output controller 120 or output devices 140. Service negotiation process 514 may include one or more of the following:

10016630-1001

(1) The user may be provided with information on the available output devices 140.

(2) The user may select a desired output service according to the information provided.

The output manager 308 residing in the information apparatus 100 may participate in, coordinate and manage the service negotiation process 514, together with the output controller 120.

**[00129]** In step 600, the user may be provided at information apparatus 100 with information on some or all available output devices 140 identified in the discovery process 502. Information being provided here may include one or more of the following:

- Total number of available output devices discovered.
- Model and make of each available output device.
- Type and capability of the output device.
- Type of service and quality of service provided by each available output device.
- Conditions under which the services are provided. For example, if service charge or subscription fee is required and the acceptable payment method.
- Whether the use of each output device is restricted to certain groups.
- The components or information, if any, need to be uploaded to the information apparatus to enable output to the selected output devices.

**[00130]** Information collected during discovery process 502 may be provided to the user through a user interface on the information apparatus 100, such as a GUI or voice or another interface.

**[00131]** In step 602, based on some or all of the information provided, the user may choose one or more output devices to take the print request or print job. On the other hand, the user may choose to decline the output service offered, and choose not to output the document or content to any of those devices. In this case, alternatives may be provided to the user as previously described with reference to step 508 (Fig. 508).

**[00132]** Authentication step 604 is optional, but may be necessary if, for example, the use of an output device 140 is restricted to a group of users. In this case, the user may have to provide authentication information to identify him/herself as part of the authorized group to use the service. Examples of authentication methods may include a user's name, password, personal identification number (PIN), ID number, signatures, security keys (physical or digital), biometric, fingerprint, voice, etc. ID number or IP address of the information apparatus 100 may also be used as authentication information. Such authentication information may be provided by the user manually or detected automatically by the output controller 120 or output device 140.

**[00133]** It should be recognized that Fig 6 only provides one embodiment of the service negotiation process 514 of present invention. A service negotiation process 514 having fewer or additional steps may also be possible. As an example, step 604 (authentication) may not be necessary if the use of an output device 140 is not restricted. As another example, one or more steps, such as credit verification, payment, deposit, escrow, and receipt processing may be added to the negotiation process 514 if a service charge is involved. All these and other possible variations, however, shall not depart from the spirit and scope of present invention.

**[00134]** FIG 7 illustrates different processing procedures that may be implemented as part of output or print job processing step 518. For this description, output device 140 is referred to as a printer and output data is referred to as print data. It should be understood that a printer is only one embodiment of an output device. Display, projection and audio devices are also output devices within the scope of present invention. In no way should the use of printer, in place of output device, and print data, in place of output data, be viewed as restricting the scope to a printing device in the following discussion.

10016630-10101

**[00135]** In step 700, output manager 308 processes an input digital document or content 701 and converts it to print data 703 that can be transmitted to output controller 120, combined controller (120C, 120F) or directly to printer engine 908. Rasterization, interpretation, encoding, decoding, etc. may take place in this step. In one implementation, print data 703 may utilize an intermediate format or language to facilitate the transmission of the digital document or content 701 to the output controller 120. This intermediate format or language may be a predefined input format or language that the output controller 120 can understand. It is also possible in this step 700 that the output manager 308 simply passes the digital document 701 in its original format to the output controller 120 with little processing. In still another exemplary implementation, the output manager 308 may process (e.g. raster image process) the digital content or document 701 into a final print data or output data format compatible to the input requirements of the output device 140. Encryption techniques may be used to assure the security of communication between information apparatus 100 and output devices 140.

**[00136]** As indicated above, output data or print data (in case of a printer) 703 generated by step 700 may be in various formats. In one embodiment, the print data 703 may be in a final format suitable for sending directly to the printer engine. In this case, the output controller 120 does little or no processing. One exemplary use of this configuration can be implemented in a low cost inkjet printer that does not have a printer controller. In this example, the print data 703 may be in a compressed 1 bit or more CMYK data format that can be sent directly to the printer engine 908 for final output as shown in step 702. Or print data 703 can be sent to the output controller 120 for possible optional features such as buffering, queuing, and print job management as shown in step 704.

10016630-110101

**[00137]** In another embodiment, output manager 308 may generate print data 703 in an intermediate format, language, or instruction that requires further processing. Examples of such format, language or instruction may include, without limitations, Page description languages (e.g. PostScript, PCL), metafiles (e.g. EMF), markup languages (e.g. XML, HTML), image or graphics formats (TIFF, GIF, PNG, JPEG compressed), among others. The intermediate format may also include proprietary solutions in page description languages, markup languages, metafiles, image and graphics formats, encoding and decoding, compression and decompression. The above-mentioned intermediate formats may be used independently, or in combinations.

**[00138]** As described above with reference to Fig. 4, output controller 120 may be implemented in a variety of ways. Different implementations of output controller 120 may impact the processing procedures described here with reference to Fig. 7.

**[00139]** As shown in step 702, the print data 703 is sent directly to the printer or display engine 908 for final output with little or no further processing from output controller 120 or a printer controller 910. In this case, the print data or output data 703 is preferably to be in a final format or language that can be understood by the printer or display engine 908 that is part of the selected output devices or devices 140.

**[00140]** Alternatively, as shown in step 704, print data 703 is sent to an output controller 120 connected to or installed within a selected printer or output device 140. In one embodiment, the print data 703 may be in a final format or language that can be understood and processed by the printer or display engine 908. In this case, the output controller may buffer the input print data 703 before passing it to the printer or display engine 908 with little or no further processing. In another embodiment, print data 703 maybe in an intermediate format or language not understood by the printer 900 or output engine 908. Therefore, the output controller 120 may process the print data and convert it into a final

1001630-110101



format, language or instruction that can be sent to the printer engine (or other output engine, in the case of other type of output device).

**[00141]** Regardless of the format of input print data 703, the output controller 120 may perform additional management functions such as payment processing, service verification, authentication, print job management, queuing, spooling, quality of service among others.

**[00142]** As illustrated in step 706 and 708, the print data 703 may be processed by both output controller 120 (in step 706) and printer controller 910 (in step 708) separately before being sent to the printer engine 908 for final output. For example, some printers, such as high-speed laser printers, already include a printer controller 910 that has the capability of processing certain types of input such as PostScript, PCL, and XML, among others. In this case, the output controller 120 may process or convert the print data 703 into the input 707 required by the printer controller 910. Alternatively, in a different implementation, if the print data 703 is already in one of the formats or languages that are compatible and can be understood by the printer controller 910, the output controller 120 may simply buffer the print data 703 and sent it directly to the printer controller 910 as needed with little or no processing. In step 708, the printer controller 910 may perform functions such as interpretation, decoding, and raster image processing operations on the input print data 707 (such as a page description language) if such operations were not already performed by output controller 120 or by output manager 308. The processed result of the printer controller 910 may be sent to the printer engine 908 for final output.

**[00143]** As illustrated in step 710, a single combined controller 120F, 120C (as illustrated in Fig. 4C and 4F) that combines the functionalities of output controller 120 and a printer controller 910 may process the print data. Applications that provide feature sets for each controller may share the same resources, such as processor, memory space, storage unit, etc. Under this configuration, the combined controller (120C, 120F)

10016330-10101

may further process the print data 703 in step 710 if the print data 703 is not already in the format or language required by printer engine. The combined controller (120C, 120F) may need to perform interpretation, decoding and rasterization operations in addition to converting the input print data 703 (in one or more of intermediate formats described previously) to the required format, language or instruction of the printer engine 908 for output. If the print data 703 is already in the final format, language or instruction understood by the printer engine 908, the combined controller (120C, 120F) might simply buffer the print data 703 and send it as needed to the printer engine 908 for output.

**[00144]** Similarly, other than processing print data, the output controller 120 and the combined controller (120C, 120F) may perform functions such as payment processing (if service charge applies); print job management, queuing, spooling, etc.

**[00145]** Figs. 8A -8E show a series of exemplary graphical user interfaces (GUIs) rendered at different times on a display screen 800 of an information apparatus 100 to illustrate one implementation of the output process 501 described with reference to Fig. 5. The user interfaces shown can be implemented in an information apparatus 100 with or without an operating system. The user interfaces illustrated here are only one example, and other user interfaces may also be implemented within the scope of the present invention.

**[00146]** Referring to Fig. 8A, a user receives an email on an information apparatus 100 using an email application. Functions available to the user are displayed at the bottom of the screen 800. Printing is one of the functions displayed by the "print" function control or icon 802. The user may invoke such functions by selecting the icons displayed. The user can make the selection by using, for example, any one or combination of a keyboard, keypad, mouse, stylus, soft keys, push buttons, software command, touch sensitive screen, etc. The user may also make such

1001630-10101

**[00150]** After the user has successfully selected the desired output device or devices in the service negotiation process 514, the information apparatus 100 may communicate with output controller 120 to identify

data or software components, if any, which need to be uploaded to the information apparatus 100 to enable printing. The output manager 308 residing in the information apparatus 100 may participate in, coordinate and manage the communications between applications in the information apparatus 100 and the output controller 120.

**[00151]** In this particular example, as illustrated in Fig. 8C, the communication between the information apparatus 100 and the output controller 120 has identified that a printer driver needs to be uploaded as displayed on screen 820. Options are provided to the user as whether or not to continue the process. This information and selection are optional as the process may be implemented transparently to the user. Other optional information that the information apparatus 100 may provide to the user includes, for example, the estimation of time needed for the uploading.

**[00152]** The above description gives an example of a printer driver component that may need to be uploaded. Other components and combinations may also be possible; this may include for example a partial print driver, data, software components, or a user interface. A partial print driver or component may for example include only device dependent software components and data specific to an output device. Device independent code and data of the device driver may have been previously installed in the information apparatus 100 and therefore need not be uploaded. It is also possible that no components need to be uploaded because all the components needed for printing to the selected output device 140 may already reside in the information apparatus 100. Assuming in this example that the user agrees to upload the driver, Fig. 8D illustrates an example of an interface when synchronization process 516 proceeds.

**[00153]** When the synchronization process 516 is completed, screen 840 is displayed to the user as shown in Fig. 8E. The user may use screen 840 to select a preference such as page range, number of copies, number of

1001630-10101

cards per page, color or grayscale, page layout parameters, etc.

Selecting "property" control or button 842, the user may be provided with more device-dependent options such as color and tone adjustments, halftoning parameters, duplexing, resolution, watermarking, etc.

**[00154]** Steps 518 and 520 can proceed after the user finishes selecting any preferences. The user, after selecting the confirmation control or "OK" button 844 shown in Fig. 8E, waits to obtain the printout or other output from the printer or printers 140 selected. Fig. 8F illustrates an interface for the printing process of steps 518 and 520.

**[00155]** After the print job has been successfully finished, the user may be provided with an option to keep or delete the printer driver or software components or synchronized device dependent data, which was uploaded to information apparatus 100 in the earlier steps. This option may be beneficial to those mobile information apparatuses 100 with limited memory space or a casual user of that particular output device 140. A user who frequently uses a selected printer may wish to keep the printer driver, software components, or data installed so that synchronization for uploading components may be skipped when the printer is used again.

**[00156]** Figs. 8A – 8E illustrate only one example of the output process shown in Fig. 5. While the above descriptions contain much specificity, these should not be construed as limitations on the scope of the invention, but rather as exemplary embodiment thereof. Other user interfaces and embodiments may also be implemented. The process may also be implemented with more or fewer steps. As an example, security verification, authentication, and payment processing may be added to steps illustrated in Figs. 8A – 8E. As another example, the step illustrated by Fig. 8C may be skipped as the loading process proceeds automatically when the user selects a desired printer or printers. These and other possible variations do not however depart from the scope of present invention.

10016630-110101

[00157] Accordingly, the scope of the invention should be determined not by the embodiments illustrated, but by the appended claims and their legal equivalents.

[00158] Having described and illustrated the principles of our invention with reference to an illustrated embodiment, it will be recognized that the illustrated embodiment can be modified in arrangement and detail without departing from such principles. In view of the many possible embodiments to which the principles of our invention may be applied, it should be recognized that the detailed embodiments are illustrative only and should not be taken as limiting the scope of our invention. Rather, I claim as my invention all such embodiments as may come within the scope and spirit of the following claims and equivalents thereto.

1001630-110101